RMF4DSR

A Risk Management Framework for Design Science Research

Pries-Heje, Jan; Venable, John; Baskerville, Richard

Published in:
Scandinavian Journal of Information Systems

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Pries-Heje, J., Venable, J., & Baskerville, R. (2014). RMF4DSR: A Risk Management Framework for Design Science Research. *Scandinavian Journal of Information Systems*, *26*(1), 57-82.
http://iris.cs.aau.dk/tl_files/volumes/volume26/Pries-Heje-etal-26-1.pdf

RMF4DSR: A Risk Management Framework for Design Science Research

Jan Pries-Heje
Roskilde University, Denmark
janph@ruc.dk

John Venable
Curtin University of Technology, Western Australia
John.Venable@cbs.curtin.edu.au

Richard Baskerville
Georgia State University, USA
baskerville@acm.org

Abstract. Design science research (DSR) is a complex form of research that combines very heterogeneous activities requiring different skills. Information systems and technologies are complex entities that have more elaborate areas of risk to manage. Research to invent and evaluate new IT artefacts introduces new areas of risk As yet, there is little experience with managing risk in DSR or even identification of types of risks to be managed. This paper analyses DSR research activities and elaborates known principles and practices of risk management and applies them to DSR to develop a new framework (RMF4DSR) for identifying, assessing, prioritizing, and treating potential risks inherent to DSR. Potential users of the framework include experienced and novice DSR researchers. The framework classifies six potential risk areas and enumerates specific key risks within each area. It includes risk assessment and treatment models. The paper applies the framework to a recent DSR case study to provide initial evidence of its value and feasibility and further evaluates the RMF4DSR framework by asking researchers to apply it in a workshop and surveying the participants' opinions about the utility of the framework.

Keywords: Risk management, design science research, framework.

Accepting editor: Jacob Nørbjerg

1 Introduction

A risk is a potential problem that can have undesirable, possibly catastrophic consequences if the risk materializes. Like most activities in life, conducting research has risks. While design science research (DSR) in the IS field is subject to the usual risks of other forms of research, its novelty, new kinds and combinations of heterogeneous research activities, and its potential complexity add extra and important risks not present in other forms of research.

There is in fact very little discussion of risk in relation to how to do research. A systematic literature search has identified very few authors on this topic. None of the major textbooks on research methods (e.g., Booth et al 2008; Burns 2000; Creswell 2013 includes risk in its index or even mentions risks as a separate topic. One major IS research method textbook (Avison and Pries-Heje 2005) discusses risk in the supervisor-student relationship (Avison and Pries-Heje 2005: chapter 3, table 3.1) and mentions risk three times in chapter 6 on industry-research relationships. If risk is treated at all, it is treated implicitly, with guidelines for activities that may include areas to watch out for (potential risks) and activities that are designed to eliminate or reduce the likelihood of specific risks. However, a literature that explicitly, holistically, and systematically addresses management of risk in research is largely missing, with the important exception of literature concerning research ethics and the reduction of risks to research participants, especially in medicine and other areas of potentially invasive research. The purpose of this paper is to fill this gap for DSR.

The objective of DSR is to develop new solutions to problems in the form of technological artefacts (Hevner et al. 2004; March and Smith 1995; Simon 1996) and design theories, which formalise knowledge about the artefacts and their utility to solve problems and make improvements (Gregor 2006; Gregor and Jones 2007; Venable 2006b; Walls et al. 1992). As shown in figure 1, DSR engages in three cycles: a relevance cycle (which ensures that DSR outcomes are relevant to business needs and appropriately disseminated), an artefact design cycle, and a rigor cycle (which ensures that DSR outcomes, especially new knowledge, are reliable and well substantiated), The Design Cycle is unique to DSR in that it invents a new technology or artefact(s); other research paradigms study existing reality rather than inventing new ways to improve reality.

In terms of risk, DSR is different from other research paradigms. Each of the cycles in figure 1 has its own peculiar risks. The artefact design cycle and its risks are missing from empirical behavioural and natural science forms of research. The consequences of the design cycle for the other cycles (e.g., the introduction of new technologies to practice or the addition of design theories/knowledge to the body of knowledge) and its risks are also missing. Therefore, DSR faces substantially different kinds of risks than in other forms of research. DSR researchers also face many of the risks that are common to more traditional empirical or theoretical research. In relation to the design cycle, DSR researchers also commonly face risks more often associated with software development (or other product development) projects. While software or other technology developers commonly face relevance and design risks, DSR adds rigor risks because research activities are concerned with the reliability and validity of new knowledge. While empirical researchers face rigor risks, DSR also faces relevance and design risks peculiar to DSR, such as solving the wrong problem or attempting to design a new technology in a way that is technically or organisationally infeasible.

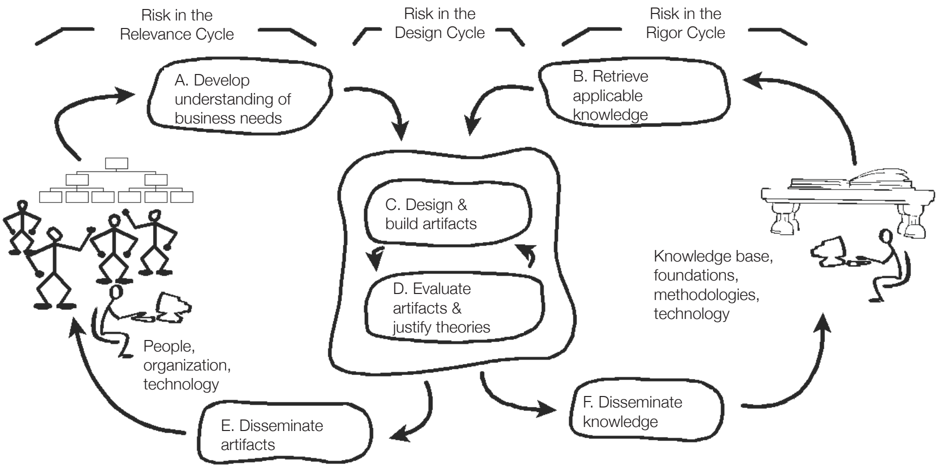

Figure 1. Risk cycles (adapted from (Hevner 2007; Hevner et al. 2004)

The purpose of this paper is to present the results of research to develop and evaluate a framework and method (which we have named the Risk Management Framework for Design Science Research, abbreviated as RMF4DSR) for DSR researchers to use to improve their ability to manage risk in DSR. To do so, we analyse, identify and describe what sorts of risks there are in undertaking DSR in IS, what causes there are for such risks, and what IS DSR researchers might do in order to assess, prioritize, and reduce or mitigate the risks identified. We analysed and identified DSR risks based on the six activities (A-F) in the above framework to develop a checklist of different DSR risks. We then developed a four-step method that applies standard risk management practices that address risk in a holistic way, but specifically addressing the risks inherent in DSR. Figure 2 below gives a preview of the RMF4DSR method, as described in full in this paper. The DSR risk management framework presented in this paper consists of six elements:

- A three-cycle, six-activity DSR risks model (figure 1).
- A four-activity risk management process (figure 2).
- Six risk-identification-class checklists. The six classes are based on of the six activities in figure 1, denoted by letters A to F. The checklists are to be used to identify risks in the specific project.
- Risk consequence and probability assessment scales, which are to be used to assess each and every one of the identified risks.
- A risk prioritisation matrix. Based on the potential consequence and probability for each risk, you place each risk in a matrix. The result of placing all identified risks are three groups of risks; High, Medium and Low.
- A four-category risk treatment model to be used to decide the treatment for each risk.

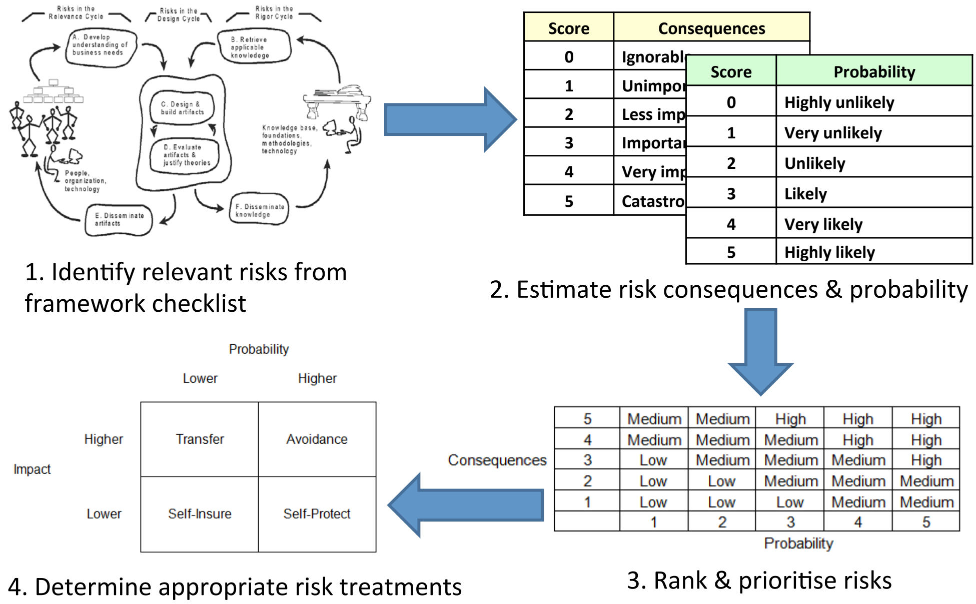

Figure 2. Overview of the Risk Management Framework for Design Science Research (RMF4DSR) Method.

RMF4DSR is intended to be used by DSR researchers early in a DSR project and possibly periodically reviewed during a DSR project. Its careful application will be most helpful for large and lengthy DSR projects, as well as those that develop safety critical artefacts. It will also be most helpful for novice DSR practitioners. However, lightweight application of the framework and checklists should also be valuable for even small DSR projects.

We evaluated (and refined) RMF4DSR through applying it in an existing research project as well as teaching the method to researchers and students and seeking feedback on it. This paper describes the rationale, form, and evaluation of RMF4DSR.

The next section of this paper reviews extant principles of risk management. Section 3 analyses the DSR process to determine risks inherent in DSR and develop a checklist of DSR risks. Section 4 then develops an approach for analysing the importance and relevance of the risks identified in section 3 for a particular DSR project. Section 5 then develops a framework for determining appropriate treatments to address identified relevant and important risks. Section 6 describes and evaluation of the developed RMF4DSR framework by applying it longitudinally over 3 years to a recent DSR project. Section 7 then describes the further evaluation of RMF4DSR in several workshops with many participants. Finally section 8 concludes the paper.

2 Principles of risk management

A risk can be defined as a potential problem that would be detrimental to a project's (DSR or otherwise) success should it materialize. This may lead to a design with incorrect or inadequate operation, rework, implementation difficulty, delay or uncertainty (Boehm and Papaccio 1991). Risk management is the identification of and response to potential problems with sufficient lead time to avoid a crisis. Thus risk management is proactive.

One interesting generalization in the extant literature on IS risks is that the higher the "innovativeness the IT represents to the organization, the higher the risks being undergone" (Willcocks and Margetts 1994). Since all research requires a new contribution, DSR thus requires something new related to the artefact design. Applying the argument that innovativeness increases risk, any DSR undertaking will have high risk.

Because DSR risks are more elaborate than just IT project risks or just empirical research risks, their analysis and management requires a broader framework. To create such an elaborate framework, we integrate general principles of risk management (taken from the IT security literature, e.g., Jones and Ashenden 2005) with principles from IT project risk management (taken from the software engineering field e.g., Boehm and Papaccio 1991; and from project management, e.g., Kerzner 2013), together with principles from IS research risk management (taken from the action research field, e.g., Avison et al. 2001).

Research on risk management has focused on specific issues such as quantifying risks and treating them. Lyytinen et al. (1998) discuss and categorise four different risk management approaches: software risk approaches, (e.g., Boehm 1991), portfolio risk approaches (e.g., McFarlan 1982), implementation risk approaches (e.g., Alter and Ginzberg 1978), and contingency (requirements) risk approaches (e.g., Davis 1982). Our framework would be closest to the software risk management approach from Boehm (1991) in that it attempts to balance a number of diverse risk categories. An alternative that could have been considered is the portfolio approach by McFarlan et al. (1982) where eight different risk management strategies are identified. However, that approach is best suited when you have many parallel activities. For a single researcher it is not as relevant to apply portfolio thinking, but at the university level with many researchers doing DSR, it may be very relevant. Another approach categorised in Lyytinen et al. (1998) is by Alter and Gintzberg (1978), where each risk item in a long list has a proposed set of resolution techniques. This could certainly be a possibility for our work. We could list resolution techniques for each risk item. However, we fear that this may limit the generative process of thinking through what should be done about each risk in the specific context. Hence we prefer—and believe it to be a better approach—to let the user (DSR researcher) decide the Risk Treatment within broader categories of treatments.

The IT security risk management literature is more closely attuned to general organizational risk management. It compels the use of a risk management framework because otherwise major risks may be overlooked or poorly understood and consequently go unnoticed and untreated (Coles and Moulton 2003). It is difficult to enumerate beforehand all possible risks to all possible DSR projects. We must be satisfied with a comprehensive framework to guide risk management. This motivates the need for an overall framework organization such as that in figure 1. At a lower level, it also offers four completely different approaches to risk treatments. Because this

	Clients	Actors	Transformation	World view	Owner	Environment
IS development/ software eng. project	Users	IS developers	From requirements to IS	Need for improvement and innovation	Management in the organisation	Market or government constraints
Empirical (descriptive research)	Public	Researcher and eventual participants	From lack of having knowledge, type I-IV	Need for more knowledge to predict	Research org./ Univ/Funding agency	Intl. research community constraints
Action research (AR)	Public & users	Researcher & user(s)	From problem to solution and knowledge, type I-IV	Research can contribute solving problems	Research & Mgmt in organisation	Intl. research & Market/gov
AR-oriented Design Science Research	Public & Users	Researcher & user(s)	From problem to solution and knowledge, type I-IV	Design can contribute solving problems	Research & Mgmt in organisation	Intl. research & Market/gov
Non-AR Design Science Research	Public	Researcher and eventual participants	Generic prob. to artifact & how-to knowledge, type V	Design can solve generic problems	Research org./ Univ/Funding agency	Intl. research community constraints

Figure 3. CATWOE for five research approaches showing the major differences

diversity is needed for an elaborate research approach like DSR, we adopted this risk treatment framework (discussed below).

From this field, we also adapted a systematic risk management process with four main activities: Identify, Analyse, Treat, and Monitor (Baskerville 2008). Identifying risk is an organized, thorough approach to seek out the real risks. Analysing risk determines the probabilities and the consequences of these risks. Treatment is deciding how to handle or minimize the risks. Monitoring is tracking risk status and changes.

Some authors include other activities than the above four. Boehm and Papaccio (1991) adds planning and further splits and separates prioritization from planning.

The IT project risk management literature is more closely attuned to the activities in the relevance and design cycles of DSR (figure 1). It adds principles such as quantifying risks, distinguishing uncertainty, and creating a sound portfolio based on risk thinking (McFarlan 1981). This body of knowledge is particularly helpful in identifying and monitoring DSR risks in the design cycle.

The IS research risk management literature is more closely attuned to the risks inherent in the power and politics of organizations and disciplines. Risk is also associated with the structures of power and control (Avison, et al. 2001; Marcus 1983). Research actions bear a certain degree of risk to the organization and the research participants. Research regarding design activities may fail when they lack the authority required to undertake the risks involved. Stakeholders become important risk elements because DSR may involve potential risks and benefits to many.

In figure 3 we have illustrated this point using the CATWOE technique (Checkland 1981; Checkland and Poulter 2006) to analyse and compare development practice as well as different forms of research, including traditional empirical research, action research (AR), AR-oriented DSR (e.g., Sein et al. 2011), and non-AR DSR (cf. Iivari and Venable 2009, who contrast three different forms of AR and DSR). The first two columns ("Client" and "Actor") and the fifth column ("Owner") in figure 3 illustrate that DSR, whether Action Research (AR)- oriented or

Non-AR, have many diverse stakeholders. Referring back to figure 1, the relevance cycle may have User and Management whereas the rigor cycle may have the author/researcher and public as stakeholders. And for the artefact design cycle they are all stakeholders. Another interesting thing that can be seen from figure 3 is that AR-oriented DSR and non-AR oriented DSR mainly vary in that when combining DSR with AR, you have the users and the management of the participating organization as extra stakeholders and in that you have to develop the particular instance in much more detail when you are doing AR-oriented DSR.

In figure 3 the underlying motivation for this paper also clearly comes forward with some interesting variation. For example it is significantly different developing for innovation and/or improvement in an organization than it is designing for the public and serving the international research community; stakeholders vary, the transformation is different, the worldview is totally different, and the environmental constraints are different—all things that clearly will lead to another set of risk factors when comparing traditional IS development with DSR. A key difference between Software Engineering (practice) and research is that research transformations produce knowledge/theory. In the case of empirical behavioural or natural science, research produces theory types I-IV, i.e., descriptive, explanatory, predictive, or explanatory and predictive (Gregor 2006). In the case of DSR, type V theory (theory for design and action) is produced (Gregor 2006).

3 Risk identification checklist for DSR

There are two main ways to identify risks. One is to gather a number of key stakeholders and brainstorm "What could go wrong in this DSR?" The second is to take a checklist of things that could go wrong or have gone wrong in other projects (i.e., potential risks) and ask one by one "Could this go wrong in this DSR project?" Checklists help ensure that you do not overlook a key risk or area of risk. Brainstorming helps ensure you do not overlook a risk that doesn't fit into a checklist. Of course, one can do both.

We develop such a checklist of potential risks by using the six major activities in three cycle DSR model as risk classes for identifying DSR risks. These six classes are enumerated A-F in figure 1. This model is particularly useful for risk identification because it facilitates an analysis of potential failures in each of the detailed, possible activities in each risk class. It also helps identify and separate risks to researchers from risks to the research subjects, consumers, or other stakeholders. The six classes are:

A. Business Needs: Identify, select, and develop understanding of business needs (problems and requirements) to address in the research (including problem analysis and choice)

B. Grounding: Search for, identify, and comprehend applicable knowledge (retrieved from the body of recorded human knowledge)

C. Build design artefacts: including instantiations, and develop design theories (hypothetical solutions to business needs or problems and theories about them)

D. Evaluate design artefacts and justify design theories or knowledge

E. Artefact dissemination and use: Disseminate new artefacs and design theories or knowledge to people and organisations for use in practice to address business need(s)

F. Knowledge additions: Disseminate new design artefacts and design theories or knowledge to the practical settings to resolve of improve upon business needs and problems

By further analysing the exact activities within each class, we can develop risk checklist items in each. We present the results of our analyses in sections 3.1-3.6.

3.1 Business needs (A)

Design Science Research is oriented toward problem solving (March and Smith 1995; Simon 1996; Walls et al. 1992) to meet business needs. Developing one's understanding and formulating a definition of the problem to be solved are important parts of problem solving (of which DSR is a special case). An analysis of this activity identifies the following potential risks for DSR (numbered for later reference).

A-1. Selection of a problem that lacks significance for any stakeholder

A-2. Difficulty getting information about the problem and the context

A-3. Different and even conflicting stakeholder interests (some of which may not be surfaced)

A-4. Poor understanding of the problem to be solved

A-5. Solving the wrong problem, i.e., a problem that isn't a main contributor to undesirable outcomes that motivate the problem solving

A-6. Poor/vague definition/statement of problem to be solved, with potential misunderstanding by others

A-7. Inappropriate choice or definition of a problem according to a solution at hand

A-8. Inappropriate formulation of the problem

3.2 Grounding (B)

DSR researchers should draw upon the extant body of knowledge from research and practice in order to facilitate and enhance the development of problem understandings and the formulation of potential solutions. All of the risks described in section 3.1 (group A) above also apply here, but other potential risks are also inherent. These are identified and described below.

B-1. Ignorance or lack of knowledge of existing research relevant to the problem understanding and over-reliance on personal experience with or imagination of the problem

B-2. Ignorance or lack of knowledge of existing design science research into solution technologies for solving the problem, i.e., lack of knowledge of the state of the art

B-3. Ignorance or lack of knowledge of existing relevant natural and behavioural science research forming kernel theories for understanding or solving the problem

3.3 Build design artefacts (C)

In this activity, a hypothetical (untried) solution is developed based on the ideas for a solution hypothesized by the DSR researcher(s). An instantiation is often constructed, although this may not be an outcome of the research. Whether instantiated or not, a solution is hypothetical until tried in practice.

C-1. Development of a conjectural (un-instantiated) solution which cannot be instantiated (built or made real)

C-2. Development of a hypothetical (untried) solution which is ineffective in solving the problem, i.e., the artefact doesn't work doesn't work well in real situations with various socio-technical complications

C-3. Development of a hypothetical (untried) solution which is inefficient in solving the problem, i.e., requiring overly high resource costs

C-4. Development of a hypothetical (untried) solution which is inefficacious in solving the problem, i.e., the artefact isn't really the cause of an improvement observed during evaluation

C-5. Development of a hypothetical (untried) solution which cannot be taught to or understood by those who are intended to use it, e.g., overly complex or inelegant

C-6. Development of a hypothetical (untried) solution which is difficult or impossible to get adopted by those who are intended to use it, whether for personal or political reasons

C-7. Development of a hypothetical (untried) solution which causes new problems that make the outcomes of the solution more trouble than the original problem, i.e., there are significant side effects

3.4 Evaluate (D)

The above risks of untried solutions may be reduced through justification (or possibly falsification) of the underlying IS Design Theory (ISDT (Walls et al. 1992)) and the evaluation of instantiations of the solution. However, evaluation itself carries risks of making errors, resulting in possible type I (false positive) or type II (false negative) errors (Baroudi and Orlikowski 1989; Baskerville et al. 2007). Baskerville et al. (2007) analysed potential sources of errors in testing ISDTs and evaluating instantiations in naturalistic evaluation (Venable 2006a). Such evaluation requires a long string of activities to be carried out rigorously, with the following potential risks.

D-1a. Tacit requirements (which by definition cannot be surfaced) are not dealt with when evaluating the solution technology, leading to failure of the solution technology to meet those requirements

D-1b. Failure to surface some or all of the relevant requirements leads to those requirements not being dealt with when evaluating the solution technology, leading to failure of the solution technology to meet those requirements

D-2. Incorrectly matching the articulated requirements to the meta-requirements of the ISDT lead to the testing of the IDST and evaluation of an instantiation of the meta-design in a situation for which neither should be applied

D-3. Incorrectly matching the meta-design or the design method to the meta-requirements (not following the ISDT correctly) leads to evaluation of something other than the correct solution technology or the ISDT as stated

D-4. Improper application of the meta-design or the design method (not in accordance with the ISDT) in designing an instantiation leads to evaluation of something other than the correct solution technology or the ISDT as stated

D-5. Improperly building an instantiation of the solution technology (such that it does not properly embody the meta-design) leads to evaluation of something other than the correct solution technology or the ISDT as stated

D-6. Difficulties in implementing the solution technology during naturalistic evaluation, due to such things as unforeseen complications within the business/organization, prevent the instantiation of the solution technology from successfully meeting its objectives

D-7. Success of the solution technology to meet its objectives is not obtained due to dynamic or changing requirements beyond the scope of the solution technology

D-8. Success of the solution technology to meet its objectives is not achieved due to poor change management practices

D-9. Determination of success or failure in reaching the objectives of the solution technology is error-prone or impossible due to disagreement about objectives or inability to measure

D-10. Existing organizational culture, local organizational culture differences (sub-cultures), political conflicts, etc. complicate the evaluation process or weaken the ability to make meaningful measurement of the achievement of the objectives of the solution technology

D-11. Existing organizational priorities, structures, practices, procedures, etc. complicate the evaluation process or ability to make/measure the achievement of the objectives D-III Emergence of new organizational or individual practices, structures, priorities, norms, culture, or other aspects that complicate the acceptability, workability, or efficiency of the application of the solution technology in a naturalistic setting

3.5 Artefact dissemination and use (E)

Once a new solution has been published and promoted to the public, especially if it doesn't work well or at all, but also even if it actually can work effectively, there are a number of other risks than those described above. The solution and how to apply it may be misunderstood and/or misapplied by those who would use it. The risks here are largely to practitioners and organizational managers, but failures may be evaluated as reflecting poorly on the solution and even the researcher(s) who developed it.

E-1. Implementation in practice of a solution does not work effectively, efficiently, and/or efficaciously

E-2. Misunderstanding the appropriate context for and limitations of the solution

E-3. Misunderstanding how to apply the solution

E-4. Inappropriate handling of adoption, diffusion, and organizational implementation

3.6 Knowledge additions (F)

The risks in this area are primarily to the researcher, but also to others engaged in the publication process and even other researchers and eventually the public at large. Different risks include the following.

F-1. Inability to publish or present research results

F-2. Publication of low significance research

F-3. Publication of incorrect research

F-4. Design artefacts prove too unique to disseminate

4 Risk assessment in DSR

For each of the risks identified, the next task is to decide how serious a risk we are talking about. Thus we evaluate the probability of occurrence as well as the consequences if the risk should occur. This is done for each risk.

The degree of risk can either be assessed in quantitative terms as the probability of unsatisfactory events multiplied by the loss associated with their outcome, or in qualitative terms by referring to the uncertainty surrounding the design and the magnitude of potential loss associated with project failure (inspired by Barki et al. 1993).

In some project management books it is recommended to evaluate the consequences as money lost when the risk occurs (cf. Kerzner 2013). However, this can be very difficult, so often an evaluation scale from for example 0 to 5 is used instead (or 0 to 10 as in Tiwana and Keil 2004). An example scale for scoring consequences of a risk is shown in table 1.

Score	*Consequences*
0	Ignorable
1	Unimportant
2	Less important
3	Important
4	Very important /serious
5	Catastrophic / critical

Table 1. Consequences impact scale

In the same way probability can be scored. Here one can use percentages to express likelihood that the risk will occur. Alternatively, one can use appropriate words. In table 2 we have shown another 0 to 5 scale, with appropriate wording for use in analysing the probability of DSR risks.

In traditional software development, the importance or priority of risks would be calculated by multiplying the consequence score by the probability score (U.S. Department of Commerce 1979). However, to correctly manipulate interval data, a risk matrix can be used, which is the approach we take as shown in figure 4.

It is important to base risk assessment on such a simple model when multiple stakeholders are involved. A simple model allows uninitiated stakeholders to participate in setting risk rankings with little training or difficulty. Moreover, different stakeholders are likely to assess risks differently, with such multiple rankings requiring mechanisms for combining or discussing and reaching consensus on risk assessment. A simple model reduces the complexity of discussion or calculations to combine ratings and allows other compilation techniques such as mode calculations or graphing.

Score	*Probability*
0	Highly unlikely
1	Very unlikely
2	Unlikely
3	Likely
4	Very likely
5	Highly likely

Table 2. Risk probability scale

Consequences	5	Medium	Medium	High	High	High
	4	Medium	Medium	Medium	High	High
	3	Low	Medium	Medium	Medium	High
	2	Low	Low	Medium	Medium	Medium
	1	Low	Low	Low	Medium	Medium
		1	2	3	4	5
				Probability		

Figure 4. Risk matrix

5 Risk treatment in DSR

Risk treatment falls into four major categories (Baskerville 2008; Jones and Ashenden 2005), which are discussed in more detail in sections 5.1-5.4:

1. Self-Insurance—means that you accept the consequences of the risk by dedicating resources toward future risk occurrence; thus you are actively prepared if the risk materializes
2. Self-Protection—means that you apply practices that prevent risks from having an impact
3. Transfer—means that you distribute all or part of the risk to someone else
4. Avoidance—means that you make a decision not to do something risky, thereby avoiding the risk.

These four categories of treatment are not specific to any particular risk. Each of the four forms of treatment can potentially be applied to any of the risks in the checklists in section 3.

These four categories and their relationship to the two dimensions of the seriousness of risk described in section 4 are depicted in figure 5. Frequency of risk occurrence defines one dimension in terms of lower frequency or higher frequency risks. This dimension is sometimes regarded as probability. The second dimension is developed in terms of lower impact or higher impact risks. This dimension is sometimes regarded as cost of loss. Each of the four general forms of treatment are typically recommended for one of the four quadrants defined by the two dimensions. These four forms of treatment are discussed in turn in sections 5.1-5.4.

5.1 Self-Insure

Self-insure treatments are appropriate for risks that are unlikely and low in impact (lower left in figure 5), hence they are the lowest or least important risks. The primary effect of self-insure risk treatments is to enable the project to absorb the impact of the risk occurrence. It is a mistake to

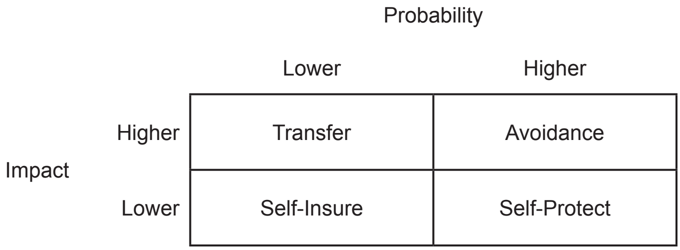

Figure 5. Risk treatments

regard this category of risk treatment as a "do-nothing" option. This treatment involves various active forms of preparedness. Typical self-insure treatments involve duplication of processing such that a failure is not catastrophic. An example of self-insurance treatments in DSR would be the production (and possibly the pursuit) of multiple designs and other contingency plans. Primarily it means that the search for solutions continues beyond the first satisfactory solution that is discovered. Having multiple designs available can overcome risks arising from poor problem definition or requirements analysis by having alternative designs available. Similarly, conducting multiple forms of evaluation may be useful should one form or episode of evaluation not work out (e.g., low survey response rate).

5.2 Self-Protect

Self-protect treatments are recommended for frequent or likely risks, but the impact of each risk is relatively low (lower right in figure 5), hence they are (generally) of medium importance. The primary effect of self-protect treatments is the reduction of impact when risks occur. Self-protect is the most common category of risk treatment and has the effect of moving the risk toward the lower left of figure 5 where self-insurance is reasonable. An example of a self-protect treatment for a DSR project would be the use of frequent or different forms of early, ex ante evaluation (Venable et al. 2012) for determining if the project is satisfying its goals. Such evaluation, a form of early warning indicator, detects anomalies and enables corrective actions. Or, if corrective action is not possible, the project can be terminated before incurring the full costs.

5.3 Transfer

Risk transfer treatments are used for risks that are high impact, but rare (upper left in figure 5), hence they are (generally) of medium importance. The primary effect of risk transfer treatments is to change the distribution of the impact of risk among different organizations. These treatments involve distributing all or part of the impact of risks to others. This is probably the second most common category of risk treatment. The classical example of this treatment used in the literature is the purchase of insurance, but insurance for a DSR project is not available. A more relevant approach for DSR would be user involvement. While not usually thought of as a risk treatment, user involvement distributes responsibility for problem definition, solution search, etc., from the designers to the users. Similarly, working in a group on a DSR project distributes the risk across multiple individuals. It also has the side effect that more creative and more appropriate solutions to risks when they occur may be discovered when more people are involved.

5.4 Avoidance

Avoidance treatments are used for risks that are highly probable and high in impact (upper right in figure 5), hence they are the highest or most important risks. The primary effect of avoidance risk treatments is to lower the probably of a risk having an important impact on the project. Avoidance may involve the basic decision not to undertake a particular DSR project. However, this decision does not necessarily mean that the research project is abandoned altogether. Example risk treatments in this quadrant could be to change the specific research topic, change the goal (e.g., problem to be solved), reduce the scope, etc. Alternatively one could shift from a DSR research design to some other form of research, such as action research or a field experiment. For example, one might conduct research to better understand the problem to be solved, possibly as a precursor to conducting DSR. Another example would be to undertake a less risky form of evaluation, one which has a lower investment in resources (lower impact) or is more likely to be successful (lower probability of risk). As a further example, risks that artefacts may be released with safety critical flaws are reduced by investing in more rigorous evaluation and testing for the most severe impacts, which reduces both the probability of the risk and its (potential) impact. Similarly, risks that the artefact evaluated might not correspond to the meta-design can be avoided or reduced by more carefully designing the artefacts in accordance with the meta-design.

5.5 Risk monitoring

Finally, after determining and applying appropriate treatments, risk monitoring should be practiced with regular follow up and asking whether anything has changed in relation to risks. Fixed intervals between continued cycles of risk identification, assessment, and treatment (e.g., at project milestones, end of phases, or at regular steering committee meetings) are recommended, particularly for large or lengthy DSR projects.

6 Illustration and evaluation of RMF4DSR: A case study

To illustrate how this risk management framework can operate in the field, we provide a case study below that relates the story of one project in which these techniques were used. SourceIT is a design science research project that received funding from the Danish Ministry of Research, Technology and Innovation from 2008-2011. The total budget was the equivalent of €4 million

(Euro) over 3 years. One university, one technology transfer organization, and three companies were participating in the project. The aim of SourceIT was to answer questions like: (1) What can you outsource?; (2) Under which internal and external conditions should you outsource? Internal conditions are in relation to the company situation and external conditions include things such as culture; (3) How can a partnership between customer and supplier optimally balance in- and out-sourcing?, (4) How can a company be innovative while at the same time optimizing sourcing?; and (5) What are the pre-conditions for optimal sourcing in relation to innovative capability?

Sourcing was defined in this project as both in- and out-sourcing, as well as the decision-making about letting customer or client organisations develop part of IT products. For example one of the participating companies develops an electronic patient journal system. One sourcing decision was how much of the system should be developed or adapted locally in the specific department at a hospital.

The SourceIT project was using a design science research approach to develop a method and tool support for sourcing decisions. It was imagined that a kind of decision support would be designed, which would help companies to make better sourcing decisions. This, however, may not be a straightforward rational decision, but rather a "wicked" problem. One possible design was therefore a design nexus instantiation (Pries-Heje and Baskerville 2008) aiming at coming up with a useful solution for the wicked sourcing problem. A wicked problem is one that has no linear solution. It was foreseen that DSR would be combined with an AR orientation, thereby covering a weakness in both research methods: namely that design science is extended with learning cycles characterising action research, thereby ensuring better learning, and that action research is extended with a formalised approach to make theory explicit.

In fact, as an important side-track for this paper, it was in the SourceIT project that the need for a better risk management tool emerged. In the project it was realised that a traditional risk analysis would not cover all the important areas where (DSR) risk could materialize. Later reflection has documented this worry as figure 3 (see earlier). In parallel with this realisation, the authors participated and contributed to the ongoing debate on DSR. At that time, design evaluation was coming forward as a hot topic, and risk and uncertainty were mentioned as related issues (Pries-Heje et al. 2008). We then prototyped the risk framework we present here in late 2007. The first version ready for full-scale use was then taken into use in January 2008.

In January 2008 and again in October 2008, project participants from the SourceIT consortium (facilitated by one of the authors of this paper) carried out an evaluation using the six areas of potential risks in DSR as an inspiring starting point. In January 2008, a list of 14 risks was generated, as shown in the bulleted list below. For each risk, the risk area(s) and the specific potential risk(s) identified in Section 3 are given in parentheses after the risk.

1. It is a risk that it may be impossible to define precisely the need for sourcing. (A-6)
2. The needs for sourcing are very different in the three participating companies. (A-3)
3. Dozens of managers have invested their soul in existing sourcing decisions; they will never admit to problems. (A-2)

4. It is hard to get access to organisations in India, to which two of the participating organisations have outsourced. (A-2)
5. There are many diverse and conflicting problem descriptions in hospitals. One of organisations parti-cipating in SourceIT wants to discuss and decide sourcing for these problems (A-3)
6. There is a vast (too much) literature and knowledge on sourcing. (B-1, B-2)
7. It may be impossible to build a Design Nexus due to lack of necessary information. (A-2, B-1, B-2, and C-1)
8. It may be impossible to build a Design Nexus because the design problem is not wicked but linear (C-1)
9. Evaluation criteria are not obvious for the appropriate choice of sourcing method. (A-2, A-3, and iteration between C & D – see figure 2)
10. It will be very difficult for many diverse stakeholders in sourcing to be involved in evaluation (D-1b, iteration between C & D – see figure 2)
11. It will take considerable time before effects of using the design artefact (sourcing nexus) can be seen (IE-1 and iteration between C & D – see figure 2)
12. It will be difficult to decide how to evaluate whether the sourcing problem is solved (E-1 and iteration between C & D – see figure 2)
13. There is no guarantee that new knowledge will be obtained. (F-1)
14. The companies participating in SourceIT are not interested in publishing results. (F-1)

The list of risks was presented and discussed in the steering committee for the SourceIT research project. In this steering committee there were representatives from the companies as well as a technology transfer organization and the participating university.

After the list was generated, the Research Manager for SourceIT (one of the authors of this paper) then used the scales from 0 to 5 (see section 4 in this paper) to evaluate the consequences and probability for each of the risks in the list. The consequence and probability ratings were then multiplied and the risks sorted by the result, which resulted in a prioritised list. In table 3, the top five risks are shown together with the treatment decided (using the treatment strategies in section 4 of this paper).

In October 2008 the same risk analysis exercise was repeated. But this time seven participants from the university and the technology transfer organization took part. The list generated this time consisted of 29 risks. Again, for each risk, the risk area(s) and the specific potential risk(s) identified in section 3 are given in parentheses after the risk.

Risk	*Prob*	*Cons*	*Treatment*
14	3	4	Transfer or Avoid: Make sure that contract for SourceIT gives researchers the right to publish—possibly keep research site anonymous
13	2	5	Transfer: Use pilots and prototypes so it becomes clear very fast what the contribution could be
3	3	3	Avoid, self-protect and/or transfer: Use many diverse problem identification techniques such as document study, observe sourcing-at-work, interview at many levels, etc.
11	4	2	Self-protect: Study effect only in projects that ends within second year of SourceIT project; leaving a full year to study long-term effects.
7	2	4	Transfer: Place considerable effort in literature study very early

Table 3. SourceIT risk analysis fragment from January 2008.

Six of the participants (plus one of the authors who facilitated the process) then used the scales from 0 to 5 (see section 4 in this paper) to evaluate the consequences and probability for each of the risks identified. Then, average consequence and probability ratings were applied using the risk matrix (figure 4), which resulted in a prioritised list (taken from upper right corner and towards lower left corner in figure 4). The top seven risks identified in the 2nd round are shown in table 4 together with the treatment decided (using the treatment strategies in section 4 of this paper).

1. We mix issues and problems between management and problem level (A)
2. It is difficult to get access to the right people in the companies (A)
3. It is impossible to build a Design Nexus due to lack of necessary information (A)
4. We don't get all the experiences from the companies activated (A-2)
5. It is hard to get access to organizations in India, to which two of the participating organizations have outsourced (A-2)
6. The real problem in the companies is different from what they tell us (A1 & A4)
7. Bad problem understanding - We don' get to the core (A-4)
8. The real motives for outsourcing in the companies remain hidden (A-5)
9. It is impossible to define precisely the need for sourcing. (A-6)
10. Too little is happening; we never get enough action in the project (A to F)
11. That we don't get enough of an overview of existing research on contracts (B-1)
12. Researchers involved don't communicate enough across 3 cases (A and B)
13. – 20. …

21. To get customer/company to invest enough resources in applying design (E)
22. It will take considerable time before effects of using the design artefact (sourcing nexus or contract model) can be seen (E)
23. – 29. …

One thing that may need some discussion is how the prioritized list was created. In January 2008 the assignment of probability and consequence was done by one person after informed discussion. This is similar to a design science researcher sitting at his desk using all available information sources to carry out a risk assessment using RMF4DSR. In October 2008 the same calculation was done by six individuals and averages were used to produce a prioritized list.

A third option would be a facilitated discussion, i.e., where six participants discuss probability and consequence until a consensus has been reached. Of the three prioritization schemes, it is our belief that it may take too much time to facilitate a discussion; thus, using averages is probably the best option.

Risk	*Prob.*	*Cons.*	*Treatment*
10	3.7	3.8	Avoidance: Would for example involve switching away from DSR. Instead it was decided to increase action in project by several means
22	3.8	3.5	Avoidance: To avoid the problems of long waiting before results of design dissemination can be seen it was decided to build a prototype of the design nexus very fast; within 6 months, thereby leaving 18 months of the project period to see results.
12	3.8	3.3	Avoidance: It was decided that all researchers should be actively involved in more than one case organization thereby avoiding lack of communication between researchers that only works on a single case.
21	3.7	3.5	Avoidance: We should consider how to build a design that do not require any significant resources to disseminate.
7	2.8	3.8	Self-protect: Place considerable effort in literature study very early. Pay careful attention to problem formulation. Use (several) problem exploration and formulation techniques
3	2.8	3.8	Self-protect: Place considerable effort in literature study very early.
4	4.3	2.3	Transfer: Collect information from a wider variety of organisations and stakeholders that are doing sourcing.

Table 4. SourceIT risk analysis fragment from October 2008

Measuring the utility of the RMF4DSR is difficult at best. A successful SourceIT project—as it was seen in the end—doesn't really provide evidence that the framework worked, as the project may have succeeded anyway. Unsuccessful projects don't clearly show that it failed either, as some risks may be difficult or impossible to estimate and others may not be amenable to anticipation. In such complex situations, success or failure may be attributed to any of a large number of potential causes. We can only rely on the perceptions of the participants. The steering committee that took part in the October 2008 assessment and followed the project to the end in 2011 saw it as very useful. In fact they would not run a project in their companies without an on-going risk assessment and treatment of the risks found. Hence what was really seen as useful was the list of risks specific to DSR. The Project Manager of SourceIT believes this list of activities to avoid or protect against risks was extremely valuable. “It is highly likely that this will make the difference between success and failure in the project”, he stated in 2010. While this last piece of evidence is anecdotal, subjective, and likely biased, combining it with the above demonstration of risk identification and prioritisation does provide some evidence of the utility of the approach.

Furthermore, the research manager emphasises that discovery of and addressing risks is the key—especially if it is reasonable to believe that the risks were only identified and addressed through the application of the framework.

Clearly though, further research and evidence was needed.

7 Further evaluation of RMF4DSR

To more rigorously evaluate RMF4DSR in practical use (naturalistic evaluation—cf. Venable 2006a; Venable, et al. 2012) the authors have conducted a number of workshops in which they have taught the framework. During the workshops, attendees applied the framework to their own research (if possible) or to a fictional piece of design science research. Workshop attendees were a mix of experienced researchers, early career researchers, and PhD students. Four workshops were conducted at four different locations. In all there were 21 attendees. During the workshops, sixteen attendees applied the framework to their own research and five applied it to a fictional DSR example.

At the conclusion of the workshop, attendees were asked to complete a short questionnaire evaluating the framework. The questionnaire included quantitative questions concerning the framework’s ease of learning, usefulness, and helpfulness for their research (only for those who applied it to their own research), as well as ratings of the likelihood that the respondent would continue to use the framework for their research (again only for those who applied it to their own research) and that they would use the framework on future DSR projects (all participants). The questionnaire also invited open comments to identify problematic areas, good and useful areas, perceived benefits, and suggestions for improving the framework. All attendees completed the questionnaires.

Table 5 presents the results of the quantitative questions. Quantitative questions asked for responses on an 11 point, zero-to-ten numeric scale, with the end points anchored on a theoretical zero point and a maximum possible point. For example, the question on usefulness of RMF4DSR asked “2. On a 0-10 scale, how easy to use is the Risk Management Framework for DSR, with zero being impossible to use and 10 being extremely useful?” As another example, the question on future use asked “On a 0-10 scale, how likely is it that you will use the Risk Management Framework for DSR on another research project in the future, with zero being no chance at all and 10 being 100% certain?" We used this scale because it conforms to a commonly understood usage ("What would you give it out of 10"), which also allows computation of averages.

From the quantitative results, the reader can see that the framework was rated very highly in ease of learning and usefulness as well as quite highly in helpfulness. The answers concerning rating of liklihood that the attendees would continue to use the framework and/or would use it on future DSR projects further indicate the utility of the framework to participants. While average ratings were generally high, not all individual ratings were high, indicating that the framework was not universally considered to have high utility for every researcher. Generally lower ratings were given by highly experienced researchers, for whom presumably their experience makes risk assessment and management a more intuitive and familiar practice.

The post-workshop questionnaire also contained a number of questions asking for open comments and suggests (e.g., 10. "What three suggestions would you make for improving the Risk Management Framework for DSR?"). Benefits and strengths of RMF4DSR cited included its systematic approach, general promotion of caution in conducting DSR, clear overview of risks, proactive and early identification of risks, and guidance on coping through risk mitigation strategies/treatment. Weaknesses/problematic areas identified by respondents included lack of clarity of explanation of some risks, inconsistencies between some risks, the difficulty in identifying risks early on in a DSR project, and difficulty in identifying appropriate treatments even if one understands the risks. The respondents made a number of useful suggestions for improving RMF4DSR, including developing a clearer overview, renaming and reorganizing the risks in the checklist, describing the risks more clearly, and providing more concrete suggestions for risk treatment in a DSR context. We have incorporated a number of the suggestions into the revised version of RMF4DSR described in this paper, including adding the four-step process (figure 2).

	Ease of Learning	*Usefulness*	*Helpfulness*	*Continue Use*	*Future Use*
Average	8.10	8.00	7.56	6.87	7.60
Minimum	3	6	3	2	3
Maximum	10	10	10	10	10
Std Dev	1.84	1.38	2.16	2.72	2.01

Table 5. Quantitative Results of RMF4DSR Evaluation Survey

As yet, the revised version of RMF4DSR presented here has not been evaluated per se, but the current version has only minor/superficial differences from previous versions, mostly pertaining to presentation and description. It is likely that RMF4DSR will continue to evolve over time as we gain more experience with it and make various small (or possibly large) improvements.

8 Discussion

One major surprise to us was how little risk is discussed in the research method literature; there is no mention of it at all in leading textbooks (Booth, et al. 2008; Burns 2000; Creswell 2013). Possibly the root cause for this gap is that the project metaphor is used very sparsely when doing research; other metaphors prevail (cf. Avison and Pries-Heje 2005: Chapter 3). There is, however, no lack of risk thinking and management in the project management literature. Kerzner (2013) for example have a section 14.17 specifically on risk management, where he divides it into 4 phases: (1) Identified; (2) Quantified; (3) Probability; and (4) Interdependency. The first three of these are very much alike to our approach.

The same four schools or categories of risk thinking (Lyytinen, et al. 1998) that we presented earlier (in the literature section 2) are found in Iversen et al (2004), which gives a specific adaptation of risk management to software process improvement. But instead of the four treatment strategies that we have, they have five: (1) Adjust Mission; (2) Adjust Strategy; (3) Mobilize; (4) Increase knowledge; and (5) Reorganise. In fact, these five risk resolution strategies can be seen as complimentary to our treatment strategies and at least four of them will often be meaningful to consider. For example, to Self-Protect one can adjust the (research) mission or reorganize (research), and when transferring risk through user involvement—the example we used earlier—you actually mobilize energy with a key stakeholder.

While evaluated very positively, the framework also has potential limitations and dangers. For example, the checklist of risks may not yet be complete and may never be complete enough to cover all possible risks. It may also need adaptation to each unique setting. Such limitations apply to most forms of checklist—that users may over-rely on the checklist and overlook risks missing from the checklist. While this is a limitation of the approach, the checklist approach does alert its users to risks that they may not otherwise have considered. Awareness of the limitation of the checklist approach seems to be the best solution. Also, as noted earlier, one can also combine checklist use with more open brainstorming.

Another potential danger is that risk management may become too large a task, particularly for small DSR projects. The larger the DSR project, the more is at risk and the more relevant the checklists and their rigorous application become. Also of course, if the artefact(s) developed is/are safety critical, careful consideration of risks is essential. Where DSR projects are conducted over a longer period of time (e.g., years) or where there is substantial change over the course of the project, ongoing review and management of risks becomes more important. However, in our experience, even on small DSR projects, thoughtfully reviewing the checklist early in the project is not onerous and may alert the DSR researcher to risks that warrant their attention and action. All of this becomes all the more essential if the DSR researcher is a novice.

Another potential danger is that the identification of a number of risks may drive researchers away from conducting DSR. In our experience, treatments can be found for nearly any kind of risk to DSR. The clarification of the nature of the risk and the identification of different ways to treat risks in DSR provided by this paper is helpful in enabling effective treatment of risks in DSR. Those few DSR projects where effective treatments cannot be found, perhaps rightly should not be conducted.

9 Conclusion

Applying general risk management tenets to activities in design science research reveals risky DSR activities, such as DSR problem choice and analysis, and many more specific and detailed risk issues within DSR activities, such as poor problem formulation, the poor articulation of requirements, an inappropriate match of requirements with the situation, improper application of a design theory, improper application of design methods, a misalignment of the artefact with the design, and complications in use.

The framework presented in this paper (figure 1 and accompanying checklist) provides a means for understanding and explaining risks in design science research. As illustrated by the case described in section 6, the framework can be further used to illuminate risks and thereby lead to extremely valuable treatments that permit practitioners to avoid or protect against risks. While there is minor overlap with risks in other research paradigms, DSR does present a number of unique risks. Therefore, classifying risks to successful conduct of DSR as completely as possible and in an integrated and holistic checklist, including relevant risks that may be familiar in other research paradigms, has significant potential value to DSR researchers.

Teaching and evaluation of RMF4DSR by researchers and students has demonstrated its value more broadly. Further, we have shown in 2-hour workshop evaluations that nearly any size DSR project, whether Action Research oriented or not, at nearly any stage, can benefit considerably from using RMF4DSR.

References

Alter, S., and Ginzberg, M., (1978). Managing uncertainty in MIS implementation. *Sloan Management Review*, (Fall): 23-31.

Avison, D., Baskerville, R., and Myers, M., (2001). Controlling action research projects. *Information Technology and People*, (14:1): 28-45.

Avison, D., and Pries-Heje, J., (2005). *Research in Information Systems* (1st ed.): Butterworth-Heinemann, Oxford; United Kingdom.

Barki, H., Rivard, S., and Talbot, J., (1993). Toward an assessment of software development risk. *Journal of Management Information Systems*, (10:2): 203-225.

Baroudi, J., and Orlikowski, W., (1989). The problem of statistical power in MIS research. *MIS Quarterly*, (13:1): 87-106.

Baskerville, R., (2008). Strategic information security risk management. In: *Information Security Policy, Processes, and Practices,* D. W. Straub, S. Goodman and R. Baskerville, (eds.), M.E. Sharpe, Armonk, New York, pp. 112-122.

Baskerville, R., Pries-Heje, J., and Venable, J. R., (2007). Soft design science research: Extending the boundaries of evaluation in design science research. In: *Proceedings of 2nd International Conference on Design Science Research in Information Systems and Technology (DESRIST '07)*, May 13-16, Pasedena, CA.

Boehm, B., and Papaccio, P., (1991). Understanding and controlling software costs. In: *Readings in Software Engineering*, T. DeMarco, (ed.), Prentice-Hall, New Jersey, pp. 31-60.

Boehm, B. W., (1991). Software risk management: Principles and practices. *IEEE Software* (January): 32-41.
Booth, W. C., Colomb, G. G., and Williams, J. M., (2008). *The Craft of Research* (3rd ed.), University of Chicago Press, Chicago.
Burns, R. B., (2000). *Introduction to Research Methods* (4th ed.), SAGE Publications, Thousand Oaks, California.
Checkland, P. B., (1981). *Systems Thinking, Systems Practice*, John Wiley, Chichester, UK.
Checkland, P. B., and Poulter, J., (2006). *Learning for Action: A short definitive account of Soft Systems Methodology and its use for Practitioners, teachers and Students*, Wiley, Chichester, UK.
Coles, R. S., and Moulton, R., (2003). Operationalizing IT risk management. *Computers & Security*, (22:6): 487-493.
Creswell, J. W., (2013). *Research Design: Qualitative, Quantitative, and Mixed Method Approaches* (4th ed.), SAGE Publications, Thousand Oaks, CA.
Davis, G. B., (1982). Strategies for Information Requirements Determination. *IBM Systems Journal*, (21:1): 4-30.
Gregor, S., (2006). The nature of theory in information systems. *MIS Quarterly*, (30:3): 611-642.
Gregor, S., and Jones, D., (2007). The anatomy of a design theory. *Journal of the Association for Information Systems*, (8:5): 312-335.
Hevner, A. R., (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, (19:2): 87-92.
Hevner, A. R., March, S. T., Park, J., and Ram, S., (2004). Design science in information systems research. *MIS Quarterly*, (28:1): 75-105.
Iivari, J., and Venable, J. R., (2009). Action research and design science research—Seemingly similar but decisively dissimilar. In: *17th European Conference on Information Systems, ECIS*, S. Newell, E. A. Whitley, N. Pouloudi, J. Wareham, and L. Mathiassen, (eds.)., Verona, Italy, pp. 2711-2723.
Iversen, J., Mathiassen, L., and Nielsen, P. A., (2004). Managing risk in software process improvement: An action research approach. *MIS Quarterly*, (28:3), 395-433.
Jones, A., and Ashenden, D., (2005). *Risk Management for Computer Security: Protecting Your Network & Information Assets*, Butterworth-Heinemann, Oxford, UK.
Kerzner, H. R. (2013). *Project Management: A Systems Approach to Planning, Scheduling and Controlling* (11th ed.), Wiley, Chichester, UK.
Lyytinen, K., Mathiassen, L., and Ropponen, J., (1998). Attention shaping and software risk—A categorical analysis of four classical risk management approahces. *Information Systems Research*, (9:3): 223-255.
March, S. T., and Smith, G. F., (1995). Design and natural science research on information technology. *Decision Support Systems*, (15:4), 251-266.
Marcus, L. (1983). Power, politics and MIS implementation. *Communications of the ACM* (26:6): 430-444.
McFarlan, F. W., (1981). Portfolio approach to information systems. *Harvard Business Review*, (59:5): 142-150.

McFarlan, F. W., (1982). Portfolio Approach to Information Systems. *Journal of Systems management*, (January): 12-19.

Pries-Heje, J., and Baskerville, R., (2008). The design theory nexus. *MIS Quarterly*, (32:4): 731-755.

Pries-Heje, J., Baskerville, R., and Venable, J. R., (2008). Strategies for design science research evaluation. In: *16th European Conference on Information Systems, ECIS,* W. Golden, T. Acton, K. Conboy, H. van der Heijden, and V. K. Tuunainen, (eds.), Galway, Ireland.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R., (2011). Action design research. *MIS Quarterly*, (35:1): 37-56.

Simon, H. A. (1996). *The Science of the Artificial* (3rd ed.), MIT Press, Cambridge, MA.

Tiwana, A., and Keil, M. (2004). The one-minute risk assessment tool. *Communications of the ACM*, (47:11): 73-77.

U.S. Department of Commerce, (1979). Guideline for automatic data processing risk analysis, guideline for automatic data processing risk analysis. National Bureau of Standards, Washington, D.C.

Venable, J. R., (2006a). A framework for design science research activities. emerging trends and challenges in information technology management. In: *Proceedings of the 2006 Information Resources Management Association Conference*, Idea Group Publishing, May 21, Washington, DC.

Venable, J. R., (2006b). The role of theory and theorising in design science research. In: *First International Conference on Design Science Research in Information Systems and Technology (DESRIST)*, Feb 24, Claremont, CA.

Venable, J. R., Pries-Heje, J., and Baskerville, R., (2012). A comprehensive framework for evaluation in design science research. In: *7th International Conference on Design Science Research in Information Systems (DESRIST 2012),* May 14,, Las Vegas, USA, Lecture Notes in Computer Science, Vol. 7286, Springer, New York, NY.

Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research*, (3:1): 36-59.

Willcocks, L., and Margetts, H. (1994). Risk assessment and information systems. *European Journal of Information Systems*, (3): 127-138.